

This is a repository copy of *Learning implicational models of universal grammar parameters*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/126850/>

Version: Published Version

Book Section:

Kazakov, Dimitar Lubomirov orcid.org/0000-0002-0637-8106, Cordoni, Guido, Algahtani, Eyad et al. (7 more authors) (2018) Learning implicational models of universal grammar parameters. In: Cuskley, C., Flaherty, M., Little, H., McCrohon, L., Ravignani, A. and Verhoef, T., (eds.) The Evolution of Language: Proceedings of the 12th International Conference (EVLANGXII). Online at <http://evolang.org/torun/proceedings/papertemplate.html?p=176> , Torun, Poland .

Reuse

This article is distributed under the terms of the Creative Commons Attribution-NoDerivs (CC BY-ND) licence. This licence allows for redistribution, commercial and non-commercial, as long as it is passed along unchanged and in whole, with credit to the original authors. More information and the full terms of the licence here: <https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

LEARNING IMPLICATIONAL MODELS OF UNIVERSAL GRAMMAR PARAMETERS

Dimitar Kazakov^{*1}, Guido Cordoni², Eyad Algahtani¹, Andrea Ceolin³, Monica A. Irimia⁴,
Shin-Sook Kim², Dimitris Michelioudakis², Nina Radkevich², Cristina Guardiano⁴, and Giuseppe
Longobardi²

^{*}Corresponding Author: dlk2@york.ac.uk

¹Department of Computer Science, University of York, York, UK

²Department of Language and Linguistic Sciences, University of York, York, UK

³Department of Linguistics, University of Pennsylvania, Philadelphia, US

⁴Dipartimento di Comunicazione ed Economia, UniMORE, Modena, Italy

Abstract

The use of parameters in the description of natural language syntax has to balance between the need to discriminate among (sometimes subtly different) languages, which can be seen as a cross-linguistic version of Chomsky’s descriptive adequacy (Chomsky, 1964), and the complexity of the acquisition task that a large number of parameters would imply, which is a problem for explanatory adequacy. Here we first present a novel approach in which machine learning is used to detect hidden dependencies in a table of parameters. The result is a dependency graph in which some of the parameters can be fully predicted from others. These findings can be then subjected to linguistic analysis, which may either refute them by providing typological counterexamples of languages not included in the original dataset, dismiss them on theoretical grounds, or uphold them as tentative empirical laws worth of further study. Machine learning is also used to explore the full sets of parameters that are sufficient to distinguish one historically established language family from others. These results provide a new type of empirical evidence about the historical adequacy of parameter theories.

1. Introduction

In historical linguistics, syntactic parameters can be used as an alternative to phonology and lexicon-based approaches in the attempt to reconstruct phylogenetic trees of languages belonging to one family. Syntactic parameters are also the only type of data that allows to approach the same task for languages belonging to different language families; indeed, by definition, languages from different families do not share lexical features (common etymologies) and the comparison of phonological features has so far been unable to suggest plausible phylogenies

for their apparent lack of sufficient time depth, and for being subject to important secondary contact effects (Longobardi & Guardiano, 2009; Creanza et al., 2015).

Distance and character-based methods (Fitch & Margoliash, 1967; Rannala & Yang, 1996) can be borrowed from population genetics to analyse syntactic parameter data. For each of these approaches, it is important to make explicit any existing dependency between parameters or otherwise the resulting models of the evolution of the languages in question will be biased, since the background assumptions on language typologies will be much looser than the actual conditions constraining possible languages (Bortolussi, Longobardi, Guardiano, & Sgarro, 2011), skewing the probabilistic estimates of historical relatedness. The database developed during the LanGeLin project¹ contains a substantial number of hand-crafted implicational rules of such nature. Here, we add to this body of work by employing Machine Learning techniques to (1) create empirical dependency models between the parameters, and (2) identify the possible groups of parameters whose values are either (2a) shared among all members of a given family, or (2b) are sufficient to separate that one family from all other languages.

The results of (1) allowed us to visualise a very complex network of *possible* dependencies, which have hitherto never been explicitly modelled as a whole. We could then use this empirical data to discuss the previously made choices of parameters and reconsider the existing implicational rules, and make changes where the alternative was deemed more appropriate by the linguistic experts. The results of (2a) have a bearing on hypotheses about the latest common ancestor of all languages in the same family. The results of (2b) can be used as an indicator about possible early evolutionary changes in the history of a given family, which led to its separation as a separate entity (clade). In all cases, the use of machine learning is meant to provide support to historical and evolutionary linguists, rather than replace their expertise and judgement.

2. The Parametric Comparison Method

Parametric theories of generative grammar focus on the problem of a formal and principled theory of grammatical diversity (Chomsky, 1981; Baker, 2001; Roberts, 2012). The basic intuition of parametric approaches is that the majority of observable syntactic differences among languages are derived from a smaller number of more abstract contrasts, drawn from a universal list of discrete, and normally binary, options, called parameters. The relation between observable patterns and the actual syntactic parameters which vary across languages is indirect: syntactic parameters are regarded as abstract differences often responsible for wider typological clusters of surface co-variation, often through an intricate deductive structure. In this sense, the concept of parametric data is not to be simplistically identified

¹LanGeLin ERC Advanced Grant project, 2012–2018.

with that of syntactic pattern: co-varying syntactic properties/patterns are in fact the empirical manifestations of such abstract cognitive structures.

Syntactic parameters are conceived as definable by Universal Grammar (UG), i.e. universally comparable, and set by each learner on the basis of her/his linguistic environment. Open parameters, or any set of more primitive concepts they can derive from (Longobardi, 2005, 2017; Lightfoot, 2017), define a variation space for biologically acquirable grammars, set (a.k.a. *closed*) parameters specify each of these grammars. Thus, the core grammar of every natural language can in principle be represented by a string of binary symbols (Clark & Roberts, 1993), each coding the value of a parameter of UG.

The Parametric Comparison Method (PCM, (Longobardi & Guardiano, 2009)) uses syntactic parameters to study historical relationships among languages. Parameters form a pervasive network of partial implications (Guardiano & Longobardi, 2005; Longobardi & Guardiano, 2009; Longobardi, Guardiano, Silvestri, Boattini, & Ceolin, 2013): one value V_i of parameter A_j , but not the other, may entail the irrelevance of parameter B, whose consequences, i.e. the corresponding surface patterns, become predictable. Under such conditions, B becomes redundant and will not be set by the learner. This rule pattern can be generalised to consider the union of several parameter-value bindings.

An important effect of the pervasiveness of parameter interdependencies is a noticeable downsizing of the space of grammatical variation: according to some preliminary experiments (Bortolussi et al., 2011), the number of possible languages generated from a given set of independent binary parameters is reduced from 10^{18} to 10^{11} when their interdependencies are taken into account. This also crucially implies a substantial reduction of the space of possible languages that a learner has to navigate through when acquiring a language.

3. Learning Dependencies between UG Parameters

Here we adopt an empirical, data-driven approach to the task of identifying parameter dependencies, which has been implemented on our database of 71 languages described through the values of 91 syntactic parameters (see Appendix A) expressing the internal syntax of nominal structures.

We set out to identify parameters whose entire range of values can be fully predicted from the values of other parameters. There is an important difference between previously published work on parameter dependencies and this paper’s contribution, which needs to be emphasised: rather than state that, for example, any language in which $P_1 = +$ will have a fully predictable value of P_2 (a fact which we encode as $P_2 = 0$), we seek parameters whose value can be deduced in *all* cases from the values of certain other parameters, e.g. as shown in the hypothetical example in Figure 1. Should such a rule prove to have universal validity, then parameter P_3 would never offer any advantage in distinguishing any two languages, yet it remains a descriptive entity entirely deducible from the other

```

if  $P_1 = +$  and  $P_2 = -$  then  $P_3 = +$ 
else  $P_3 = -$ 

```

Figure 1. Parameter dependency model example

parameters.

We process our table of dimensions ($\#lang \times \#param$) with the data mining package WEKA (*v.3.6.13*) (Hall et al., 2009). More specifically, we take the values of all parameters but one for all languages (i.e. a dataset of size $(\#lang \times \#param - 1)$), and learn a decision tree that predicts the value of the remaining parameter from the values of the other parameters. (Typically, only a few are necessary in each case.) This is repeated to produce a decision tree for each of the parameters. The machine learning algorithm used was ID3 (Quinlan, 1986). The algorithm produces a decision tree, in which each leaf corresponds to the value of the modelled parameter for the combination of parameter values listed on the way from the root to that leaf, e.g.: **if** $FGN = -$ **and** $FGP = +$ **then** $GCO = +$ (see Figure 3). Unlike some of the more sophisticated decision tree learning algorithms, such as C4.5 (Quinlan, 1993), no postprocessing of the tree learnt (such as *pruning* (Mitchell, 1997)) takes place, and the tree remains an accurate, exact reflection of the training data. If the combination of parameter values corresponding to one of the leaves of the tree is not observed in the data, the leaf contains the special label ‘null’ (see the tree predicting GCO in Figure 3). In all other cases, that is, whenever the leaf label is ‘+’, ‘-’ or ‘0’, this is supported by one or more examples (languages) in the data.

```

~~~~~
FGN:
if  $GCO = 0$  then  $FGN = +$ 
if  $GCO = +$  then  $FGN = -$ 
if  $GCO = -$  then  $FGN = -$ 
~~~~~
GCO:
if  $FGN = 0$  then  $GCO = \text{null}$  ;never occurs
if  $FGN = +$  then  $GCO = 0$ 
if  $FGN = -$  then
    if  $FGP = 0$  then  $GCO = \text{null}$ ;never occurs
    if  $FGP = +$  then  $GCO = +$ 
    if  $FGP = -$  then  $GCO = -$ 
~~~~~

```

Figure 2. Examples of decision trees for parameters FGN and GCO

The decision trees for all parameters were used to produce a dependency graph in which each vertex represents a parameter, and directed edges link the parame-

The interpretation of the graph is straightforward. For instance, looking at its top right corner, one can deduce that for any language in the dataset, it is enough to know the values of parameters *EZ3* and *PLS* in order to know the value of *EZ2*, and therefore, of *EZ1*, too. Knowing (the value of) *FVP* means one also knows *DMG* and *NSD*; if one knows both *FVP* and *DNN*, the values of *DNG*, *NSD*, *DSN*, *DMP* and *DMG* are fully predictable for the given data set. In other words, 7 parameters (*FVP*, *DNN*, *DNG*, *NSD*, *DSN*, *DMP* and *DMG*) can be reduced to just 2 without any loss of information.

Some of the rules identified by the algorithm are not new, and are already contained in the dataset, as encoded by the implicational system described in Section 1. For instance, the parameter *RHM* is encoded as 0 when *FGP* = −, as the value of *RHM* is fully predictable in those cases. When a decision tree predicting *FGP* is learned, the result is as follows: **if** *RHM* = 0 **then** *FGP* = − **else** *FGP* = +.

Even the rest of the rules learned are still just empirical findings: they may change with the addition of other examples of languages or their validity may be questioned by linguists on theoretical grounds.

Linguistic analysis of the results is ongoing, and while no part of the results has been accepted as sufficient evidence to dispose of a parameter, implication rules have been revised on the basis of the decision trees learned, as in the case of the parameter *PLS*. According to our definition, the parameter asks if in a language without grammaticalized Number, a plural marker can also appear outside a nominal phrase, marking a distributive relation between the plural subject and the constituent bearing it. (E.g. *PLS* = + for Korean, but *PLS* = − for Japanese.)

Prior to this research, there was an implication rule stating that *PLS* is neutralised (that is, its value is predictable) for all combinations of *CGO* and *FGN* values other than *CGO* = − and *FGN* = −. This rule has now been replaced with a new rule stating that *PLS* is neutralised for all combinations of values of *FGM* and *FGN*, except when *FGM* = + and *FGN* = −, and the evidence showing that the new rule is consistent with the data came from the tree learned for *PLS*.

4. Learning Language Family Descriptions

The existing parameters (see Appendix A) have been introduced in order to ensure each language in the database can be uniquely described and separated from the rest on their basis. On a more general level, one could search for the conditions that separate languages from one linguistic family from all others. This is, of course, a classical machine learning task of producing (training) a classifier, which could be used for two purposes, to classify new languages as they are added to the database or to describe the conditions separating one family from the rest. Again, a decision tree can be produced for this purpose. However, it will only contain a very small number of constraints on the parameters that is sufficient for correct

classification. Instead, we can adopt another algorithm, namely, Candidate Elimination (CE) (Mitchell, 1997) to learn *all* possible hypotheses (classifiers). This is a classical algorithm for learning in logic, which uses propositional data (i.e. of type $Param_1 = Value_1 \wedge \dots \wedge Param_n = Value_n$) and produces propositional hypotheses, each of which is a conjunction of one or several parameter-value pairs. Each of these hypotheses covers (implies) all positive examples, and does not cover any of the negative (i.e. it is consistent and complete). If no hypothesis of this form and properties can be produced, the result is an empty set of hypotheses. The set of all hypotheses is also known as the *version space* of hypotheses for the given dataset.

While such logic-based approach makes the algorithm rather sensitive to any noise (errors) in the data, here we make the assumption that at this stage of the work, our data is error-free. The output of CE consists of three parts: (1) the set of most specific hypotheses \mathcal{S} , i.e. those that cannot be made strictly more specific (by constraining yet another parameter) without becoming incomplete; (2) the set of most general hypotheses \mathcal{G} , i.e. those that cannot be made strictly more general without becoming inconsistent, and (3) the rest of the version space, made of hypotheses that are strictly more general than some hypothesis in \mathcal{S} , and strictly more specific than some other hypothesis in \mathcal{G} .

We applied CE to learning the description of two families of languages, namely, the Romance and the Indo-European (IE), in order to explore the insights it provides. Both families are well established, with the latter subsuming the former. There was a single most specific hypothesis (MSH) for each of the two families (see Table 1). All constraints for the IE family are shared with the Romance family, as expected, while the parameter constraints listed in bold face are specific to the Romance family. This distinction can help guide hypotheses about the last common ancestor of each family, thus providing insight into the evolution of the languages within each family, and the parameters that defined their divergent properties.

Looking at the set \mathcal{G} of most general hypotheses (MGHs) for each family can provide further insight in this direction. While the only MSH in \mathcal{S}_{IE} contains 29 parameters (of which 10 zeros, that is, fully predictable), there are numerous MGHs in \mathcal{G}_{IE} that make use of only 2 or 3 parameters, e.g.: (+GSC, -GAL), (-GAL, +PCA), (+FGM, +GSC, -GAL), (+FSN, -XCN, -GUN). A closer look at these parameters reveals that these are particularly useful to delineate boundaries between language families, e.g. -GAL for IE vs. +GAL for Dravidian, Semitic and Uralic languages or +XCN (Dravidian) vs. -XCN (all other families in the database).

5. Discussion

The results reported here show that applying machine learning techniques to the data can reveal previously unknown dependencies between parameters, leading

Table 1. Most specific hypotheses for the IE and Romance families

Indo-European family								
+FGP	+FGM	-FPC	-FGT	+FGN	0GCO	0PLS	+FND	+FSN
-DIN	0FGC	0DBC	-XCN	+GSC	-HMP	+AST	-GCN	0GFN
-GAL	-GUN	-GSI	-ALP	0GST	0GEI	0GNR	0STC	0PMN
+CQU	+PCA							
Romance family								
+FGP	+FGM	-FPC	-FGT	+FGN	0GCO	0PLS	+FND	+FSN
+SGE	+FGG	-CGB	+DGR	0DGP	+CGR	+NSD	-DGD	-DIN
0FGC	0DBC	-XCN	+GSC	+NOE	-HMP	+AST	+FFS	0ADI
-GCN	0GFN	-GAL	-GUN	-EZ1	-EZ2	-EZ3	+GAD	-GFO
0GFS	-GSI	-ALP	0GST	0GEI	0GNR	0STC	-GPC	0PMN
+CQU	+PCA	+PSC	-RHM	+FRC	-NRC	+NOR	0AER	+ARR
-DOR	-NOD	+NOP	+PNP	-NPP	+NOA	+NM2	0FPO	0ACM
-DOA	-NEX	-NCL	0ACL	+TDC	0TNL			

to a potentially significant reduction in the search space of possible languages. The data contain more features (i.e. parameters) than data points (i.e. languages), which can make for the generation of spurious rules. The most obvious way to counteract this, adding more languages, comes at a very high cost, as it requires well-trained linguists and an abundance of subtle though typologically wide evidence. One can also use Occam's Razor and limit the search space of possible rules by limiting the number of antecedents in the rule, e.g. to two as we did here. Yet another approach is to collect data selectively for rules of interest, as only a small number of parameters, e.g. 2–3 per language, will be needed to test each rule.

This research could have important implications for the understanding of processes underlying the faculty of language (potentially strengthening the case for UG through strengthening its adequacy as a restrictive typological model and as tool for insightful historical reconstructions), with consequences ranging from models of language acquisition to phylogenetic linguistics, where the syntactic relatedness between two languages may be more adequately measured. However, the approach requires a close collaboration between a machine learning expert, discovering empirical laws in the data, and a linguist who can test their plausibility and theoretical consequences. There is also an open theoretical computational learning challenge here presented by the need to estimate the significance of empirical findings from a given number of examples (languages) with respect to the available range of discriminative features in the dataset.

References

- Baker, M. (2001). *The atoms of language*. New York: Basic Books.
- Bortolussi, L., Longobardi, G., Guardiano, C., & Sgarro, A. (2011). How many possible languages are there? In G. Bel-Enguix, V. Dahl, & M. Jiménez-López (Eds.), *Biology, computation and linguistics*. Amsterdam: IOS.

- Chomsky, N. (1964). *Current issues in linguistic theory*. The Hague: Mouton.
- Chomsky, N. (1981). *Lectures on government and binding*. Dordrecht: Foris.
- Clark, R., & Roberts, I. (1993). A computational model of language learnability and language change. *Linguistic Inquiry*, 24, 299–345.
- Creanza, N., Ruhlen, M., Pemberton, T. J., Rosenberg, N. A., Feldman, M. W., & Ramachandran, S. (2015). A comparison of worldwide phonemic and genetic variation in human populations. *Proceedings of the National Academy of Sciences*, 112(5), 1265–1272.
- Fitch, W. M., & Margoliash, E. (1967). Construction of phylogenetic trees. *Science*, 155(3760), 279–284.
- Guardiano, C., & Longobardi, G. (2005). Parametric comparison and language taxonomy. In M. Batllori, M. L. Hernanz, C. Picallo, & F. Roca (Eds.), *Grammaticalization and parametric variation* (pp. 149–174). Oxford: OUP.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software. *ACM SIGKDD Explor. Newsl.*, 11, 149–174.
- Lightfoot, D. W. (2017). Discovering new variable properties without parameters. *Ling. Anal.*, 41. (Spec. ed.: Parameters: What are they? Where are they?)
- Longobardi, G. (2005). A minimalist program for parametric linguistics? In H. Broekhuis, N. Corver, M. Huybregts, U. Kleinhenz, & J. Koster (Eds.), *Organizing grammar: Linguistic studies*. Berlin/NY: Mouton de Gruyter.
- Longobardi, G. (2017). Principles, parameters, and schemata: A radically underspecified UG. *Ling. Anal.*, 41. (Spec. ed.: Parameters: What are they? Where are they?)
- Longobardi, G., & Guardiano, C. (2009). Evidence for syntax as a signal of historical relatedness. *Lingua*, 119(11).
- Longobardi, G., Guardiano, C., Silvestri, G., Boattini, A., & Ceolin, A. (2013). Toward a syntactic phylogeny of modern Indo-European languages. *Journal of Historical Linguistics*, 3(1), 122–152.
- Mitchell, T. (1997). *Machine learning*. MIT.
- Quinlan, R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81–106.
- Quinlan, R. (1993). *C4.5: Programs for machine learning*. San Matteo, CA: Morgan Kaufmann Publ.
- Rannala, B., & Yang, Z. (1996). Probability distribution of molecular evolutionary trees: A new method of phylogenetic inference. *Journal of Molecular Evolution*, 43(3), 304–311.
- Roberts, I. (2012). On the nature of syntactic parameters: a programme for research. In C. Galves, S. Cyrino, R. Lopes, F. Sandalo, & J. Avelar (Eds.), *Parameter theory and language change* (pp. 319–334). Oxford: OUP.

Appendix A: List of Parameters

FGP	gramm. person	GSI	grammaticalised inalienability
FGM	gramm. Case	ALP	alienable possession
FPC	gramm. perception	GST	grammaticalised Genitive
FGT	gramm. temporality	GEI	Genitive inversion
FGN	gramm. number	GNR	non-referential head marking
GCO	gramm. collective number	STC	structured cardinals
PLS	plurality spreading	GPC	gender polarity cardinals
FND	number in D	PMN	personal marking on numerals
FSN	feature spread to N	CQU	cardinal quantifiers
FNN	number on N	PCA	number spread through cardinal adjectives
SGE	semantic gender	PSC	number spread from cardinal quantifiers
FGG	gramm. gender	RHM	Head-marking on Rel
CGB	unbounded sg N	FRC	verbal relative clauses
DGR	gramm. amount	NRC	nominalised relative clause
DGP	gramm. text anaphora	NOR	NP over verbal relative clauses/ adpositional genitives
CGR	strong amount		
NSD	strong person	AER	relative extrap.
FVP	variable person	ARR	free reduced rel
DGD	gramm. distality	DOR	def on relatives
DPQ	free null partitive Q	NOD	NP over D
DCN	article-checking N	NOP	NP over non-genitive arguments
DNN	null-N-licensing art	PNP	P over complement
DIN	D-controlled infl. on N	NPP	N-raising with obl. pied-piping
FGC	gramm. classifier	NGO	N over GenO
DBC	strong classifier	NOA	N over As
XCN	conjugated nouns	NM2	N over M2 As
GSC	c-selection	NM1	N over M1 As
NOE	N over ext. arg.	EAF	fronted high As
HMP	NP-heading modifier	NON	N over numerals
AST	structured APs	FPO	feature spread to genitive postpositions
FFS	feature spread to struct. APs	ACM	class MOD
ADI	D-controlled infl. on A	DOA	def on all +N
DMP	def matching pron. poss.	NEX	gramm. expletive article
DMG	def matching genitives	NCL	clitic poss.
GCN	Poss ^o -checking N	PDC	article-checking poss.
GFN	Gen-feature spread to Poss ^o	ACL	enclitic poss. on As
GAL	Dependent Case in NP	APO	adjectival poss.
GUN	uniform Gen	WAP	wackernagel adjectival poss.
EZ1	generalized linker	AGE	adjectival Gen
EZ2	non-clausal linker	OPK	obligatory possessive with kinship nouns
EZ3	non-genitive linker	TSP	split deictic demonstratives
GAD	adpositional Gen	TSD	split demonstratives
GFO	GenO	TAD	adjectival demonstratives
PGO	partial GenO	TDC	article-checking demonstratives
GFS	GenS	TLC	Loc-checking demonstratives
GIT	Genitive-licensing iterator	TNL	NP over Loc